

REMARKS

Claims 1, 13, 14, and 23 are amended. No other claims have been amended, canceled, or added. Thus, claims 1-29 are pending.

Claim Rejections under 35 U.S.C. §102(b)

The October 24, 2006 office action (“Office Action”) rejects claims 1, 3, 10, 11, 14, 18-20, 22, 23 and 27-29 under Section 102(b) as anticipated by Bossen (U.S. Patent No. 6,058,491).

“A claim is anticipated only if **each and every element as set forth in the claim** is found, either expressly or inherently described, in a single prior art reference.” MPEP § 2131. Further, “[t]he **identical invention** must be shown in **as complete detail as is contained in the ... claim.**” *Id.* (emphasis added).

For at least the reasons stated below, the above claims are not anticipated by Bossen. Claim 1, as amended, recites:

1. A method comprising:

executing corresponding instruction threads as a leading thread and a trailing thread;

saving a processor state corresponding to execution of a selected instruction in a history buffer **in connection with retiring the selected instruction and before committing a result from the selected instruction to a destination register;**

after retiring the selected instruction and committing the result from the selected instruction to the destination register, comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread; and

restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault.

(Emphasis added).

The emphasized portion of claim 1 recites saving a processor state corresponding to execution of a selected instruction in connection with retiring the selected instruction and before committing a result from the selected instruction to a destination register. After retiring the instruction and committing the result of the instruction, execution results in the leading and trailing thread are compared to detect any fault. In summary, claim 1 recites retiring an instruction and committing the results of its execution before performing fault detection related to that instruction. Bossen fails to teach at least these limitation of claim 1 and therefore does not anticipate claim 1.

Claim 1 recites performing fault detection (i.e., comparing execution results) after an instruction retirement and its results committed to a destination register. This contrasts with other systems that do not retire instructions or commit the results of instruction execution until after fault detection. As explained by the specification, when an instruction is retired, its results are committed. “[I]nstructions that have been executed but have not yet been retired cannot commit the results to the destination register in the architectural register file until after the instructions have passed fault detection and retirement stages.”

(Specification, pp. 14-15, ¶ [0035]). Conventionally, when instructions are retired, they are committed. That is the usual meaning in the relevant art – “[R]etirement guarantees that instructions are done.” Rotenberg et al., “Trace Processors,” IEEE (1997).

When the results of an instruction are committed to a destination register in connection with instruction retirement, the state of the register is visible:

The commit unit keeps track of all the pending instructions in its reorder buffer. Because both machines use branch prediction, and instruction isn't finished until the commit unit says it is. When the branch functional unit determines

whether or not a branch was taken, it informs both the branch prediction unit, so that it can update its state machine, and the commit unit, so that it can decide the fate of pending instructions. If the prediction was accurate, **the results of the instructions after the branch are marked valid and thus can be placed in the programmer-visible registers and memory.** *Computer Organization & Design: The Hardware/Software Interface*, 2nd Edition, David A. Patterson and John L. Hennessy, Morgan Kaufmann Publishers, Inc., 1998, Section 6.9 "Real Stuff: PowerPC 604 and Pentium Pro Pipelines", pp. 517-518 (Emphasis added).

Storing processor state associated with a retiring instruction in a history buffer allows fault detection to be postponed until after instruction retirement and after a result of the instruction has been committed. "By updating the history buffer when an instruction is retired an incremental checkpoint corresponding the processor state at the time the instruction is retired is created." (Specification, pp. 15-16, ¶ [0038]). "When a retired instruction passes the fault detection stage, the corresponding entry in the history buffer can be freed." *Id.* p. 15, ¶ [0039]. If a fault is detected after instruction retirement, the history buffer can be used to recover. *See, e.g.*, Specification, Fig. 7.

Bossen describes performing fault detection prior to committing the results of an instruction – and hence before instruction retirement – and therefore does not anticipate claim 1:

When the leading processor (processor 12 in this example) reaches a checkpoint during operation, the leading processor leaves its processing results, preferably in a check/wait buffer, for the lagging processor (processor 14 in the same example) to compare. **The processing results stored in the check/wait buffer will not be committed for further processing until a match is confirmed** with the result at the same operational checkpoint from the lagging processor. **Only when there is a**

match of processing results at the same operational checkpoint from the two processors will further processing be allowed.

(Bossen, col. 3, lines 19-29) (emphasis added).

Thus, Bossen does not retire or commit the results of instructions until after fault detection – “there is a match of processing results.” This is contrary to what is recited in claim 1. Claim 1 recites retiring instructions, committing their results, and then performing the comparisons for fault detection. Applicants are not aware of any cited portion of Bossen which describes performing fault detection after instructions are retired and their results committed. Bossen therefore does not anticipate claim 1.

In rejecting the prior version of claim 1, the Office cites col. 4, lines 61-63 of Bossen as describing, “saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register.” (Office Action, p. 2). This cited portion of Bossen merely describes an “Instruction history buffer 39 [that] can hold, for example, two instructions.” (Bossen, col. 4, lines 62-63). Nothing in this cited passage describes saving the processor state when an instruction is retired as recited by amended claim 1.

The Office also cites col. 4, lines 61-66 of Bossen as describing “comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread.” (Office Action, p. 2). This cited portion of Bossen includes the cited portion described above. It also describes that if a mismatch is detected, then using retry logic to recover from the fault. However, nothing therein describes performing the fault detection after relevant instructions have been retired and their results committed.

Thus, the cited portions of Bossen fail to teach the above limitations of claim 1. Bossen therefore does not anticipate claim 1.

The remaining independent claims, 11, 14 and 23, have been amended to recite storing processor state related to execution of an instruction in a history buffer in connection with retiring an instruction and before its results are committed. These claims are amended to further recite performing fault detection after the instruction is retired and its results committed. The above argument relating to claim 1 is therefore fully applicable to independent claims 11, 14 and 21. These remaining independent claims are therefore also not anticipated by Bossen.

Dependent claims 3, 10, 18-20, 22 and 27-29 depend from one of allowable independent claims 1, 14 and 23. They are deemed to recite the limitations of their respective independent claims. Dependent claims 3, 10, 11, 18-20, 22 and 27-29 are therefore also not anticipated by Bossen. *See*, MPEP § 2143.03.

Claim Rejections under 35 U.S.C. §103(a)

Bossen in view of Mukherjee

The Office Action rejects dependent claims 2 and 21 under Section 103(a) as being unpatentable over Bossen in view of Mukherjee (2001/0034854). (Office Action, p. 6). For at least the reasons stated below, claims 2 and 21 are not rendered obvious by Bossen in view of Mukherjee.

Claims 2 and 21 depend, respectively, from allowable independent claims 1 and 14, respectively. As discussed above, Bossen does not teach or suggest all the limitations of claims 1 and 14. Further, Mukherjee is not cited as teaching or suggesting the limitations of independent claims 1 and 14. (Office Action, pp. 6-7). Independent claims 1 and 14 are

therefore not rendered obvious by the combination of Bossen and Mukherjee. Therefore, dependent claims 2 and 21 are also not rendered obvious by the combination of Bossen and Mukherjee. MPEP § 2143.03.

Bossen in view of Ando

The Office Action rejects dependent claims 4-9, 12, 13, 15-17 and 24-26 under Section 103(a) as being unpatentable over Bossen in view of Ando (U.S. Patent No. 6,519,730). (Office Action, p. 6). For at least the reasons stated below, claims 4-9, 12, 13, 15-17 and 24-26 are not rendered obvious by Bossen in view of Ando.

Claims 4-9, 12, 13, 15-17 and 24-26 depend, respectively, from one of allowable independent claims 1, 11, 14 and 23. As discussed above, Bossen does not teach or suggest all the limitations of independent claims 1, 11, 14 and 23. Further, Ando is not cited as teaching or suggesting the limitations of independent claims 1, 11, 14 and 23. (Office Action, pp. 6-7). Nor are Applicants aware of any portion of Ando that describes storing processor state relating to an instruction when the instruction is retired and the result of the instruction committed. Applicants note that at least the following portion of Ando also describes performing fault detection prior to retiring an instruction and committing the result of its execution:

If an instruction that caused an error has committed, the instruction cannot be canceled and, therefore, the error cannot be corrected; accordingly, the present invention employs the configuration in which instructions are committed by not only checking the usual instruction commit condition but also confirming that no errors have occurred.

(Ando, col. 3, lines 46-51).

Thus, independent claims 1, 11, 14 and 23 are not rendered obvious by the combination of Bossen and Ando. Therefore, dependent claims 4-9, 12, 13, 15-17 and 24-26 are also not rendered obvious by the combination of Bossen and Ando. MPEP § 2143.03.

CONCLUSION

Invitation for a Telephone Interview

The Examiner is requested to call the undersigned at (503) 439-8778 if there remains any issue with allowance of the case.

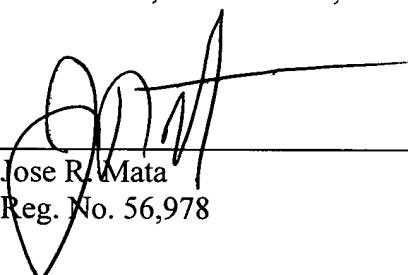
Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: January 24, 2007



Jose R. Mata
Reg. No. 56,978

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(503) 439-8778